UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/696,467	10/28/2003	Yu Kwong Ng	CSCO-7059	6905

7590 11/22/2005

WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

EXAMINER	
WALTER, CRAIG E	

ART UNIT	PAPER NUMBER
2188	

DATE MAILED: 11/22/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/696,467	NG ET AL.	
	Examiner	Art Unit	
	Craig E. Walter	2188	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 October 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☐ Claim(s) *1-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *28 October 2003* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Drawings

1. Figures 2-4 and 7a should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

2. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required:

The "computer-readable medium" as claimed in claims 21-25 is not disclosed in Applicant's specification.

Claim Objections

3. Claims 1-25 are objected to because of the following informalities:

As for claims 1, 15 and 21, the phrase "the partial key" in line 6 of claim 1 (and lines 7 and 10 of claims 15 and 21 respectively) should be changed to "a partial key".

As for claims 3, 5, 17, 19, 21 and 23, the phrase "the partial key" in line 1 of claim 3 (and lines 2, 1, 2, 10, and 1 of claims 5, 17, 19, 21 and 23 respectively) should be changed to "the partial key from the memory corresponding to the hash value" for clarity.

As for claims 9, 13, and 14, the phrase "the partial key" in line 1 of claim 9 (and line 3 and 4 of claims 13 and 14 respectively) should be changed to "the one of the plurality of keys stored in the memory" for clarity.

As for claim 7, the phrase "applies a any" in line 4 should changed to "applies any".

As for claim 13, the word "including" should be added between the word "further", and the phrase "a reverse function" in line 1 of this claim.

Claims 2, 4, 6, 8, 10-12, 16, 18, 20, 22 and 24-25 are further objected to as they depend on a previously rejected based claim.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 2 and 22 recites the limitation "wherein saved bits comprise" in line 1 (and lines 1-2 of claim 22) of the claims. There is insufficient antecedent basis for this limitation in the claims, as "saved bits" are not previously set forth in this claim, or in a claim from which they depends. The claims will be further treated on there merits based on the assumption that the partial keys stored in memory (as described in claim 1 and 22) include the claimed saved bits. Support for this assumption is provided by claim 7.

Claims 3 and 23 further limit claims 2 and 22; therefore they too are rejected as they inherit the antecedent basis deficiencies of claims 2 and 22.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 7-10 are rejected under 35 U.S.C. 102(e) as being anticipated by Brandin et al., hereinafter Brandin (US Patent 6,493,813 B1).

As for claim 7, Brandin teaches a hashing apparatus, comprising:

a memory which stores a plurality of partial keys used to determine hashing conflicts (Fig. 13a illustrates two partial keys (elements 316 and 320). Referring to Fig. 1, the memory management system (20) has three

elements including the transform generator, a controller and the memory table (26). The keys (which are split into partial keys as illustrated in Fig. 13a-b), are provided to the transform generator – col. 2, lines 54-57).

Referring to Fig. 9, each key is stored as an entry in the memory table (i.e. 10(exp)8 keys) – The transform generator determines an address and a confirmer for each key (col. 2, lines 47-48. The information determined from each of the keys (or partial keys as shown in Fig. 13a) is used to prevent the occurrence of collisions (i.e. hashing conflicts) – col. 2 lines 21-30);

a hash function block coupled to a memory that applies any polynomial to a full key and generates a hash value which is used to point to one of the plurality of partial keys stored the in memory wherein the partial keys include saved bits comprising a consecutive, sequential string of bits derived from the original key (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to partial keys in Fig 13a. – the original key (element 312) is split into partial keys, and the hash function is applied. Additionally, referring to Fig. 13a, the original key (element 312) is comprised of two partial keys (elements 316 and 320). The bits in each partial key are stored in a sequential line (based on the key length), each containing less bits than the original key – col. 7, lines 14-39);

As for claim 8, Brandin teaches the hashing apparatus of Claim 7, wherein the memory comprises a 2(exp)N hash table size (referring to Fig. 3, the store table example used (element 50) contains 16 entries (i.e. N=4)).

As for claim 9, Brandin teaches the hashing apparatus of Claim 7, wherein the partial key comprises a number of bits equal to or more than the number of bits of the original key minus the number of bits of the hash value (referring again to Fig. 13a, partial key A (element 316) is input into the LFSR to generate a hash value (transform) which is equal in size to the partial key. Since the partial key is half the original key's size, the partial key is equal to the size of the original key minus the hash value – col. 7, lines 14-39).

As for claim 10, Brandin teaches the hashing apparatus of Claim 7, wherein the hash function block comprises a linear feedback shift register (Fig. 12, element 312 illustrates the LFSR – col. 7, lines 9-11).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 11-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin as applied to claim 7 above, and in further view of Rajski et al., hereinafter Rajski (US PG Publication 2002/0016806 A1).

As for claims 11 and 12, Brandin fails to teach his LFSR as corresponding to either a Fibonacci, or a Galois version.

Rajski however teaches a method for synthesizing linear finite state machines, which includes both the Fibonacci, or a Galois versions – paragraph 0002, lines 17-20 and paragraph 0003, lines 1-4 – both types are described in his teachings.

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Rajski's method for synthesizing linear finite state machines for his own LFSRs. By doing so, Brandin would be able to more efficiently implement his LFSR with fewer levels of logic, and a lower internal fan-out of the circuitry, as taught by Rajski (paragraph 0012, lines 1-18).

7. Claims 1-4 and 15-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin in further view of Biran (US Patent 6,345,347 B1).

As for claim 1, Brandin teaches a method for hashing, comprising:

storing a plurality of partial keys in memory (Fig 13a illustrates two partial keys (elements 316 and 320). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). The keys (which are split into partial keys as illustrated in Fig. 13a-b), are provided to the transform generator – col. 2, lines 54-57. Referring to Fig. 9, each key is stored as an entry in the memory table (i.e. 10[exp]8 keys));

applying a hash function to an original key to generate a hash value, wherein said hash function comprises any polynomial (col. 2, lines

54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to partial keys in Fig 13a.);

accessing the memory according to the hash value (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49);

reading the partial key from the memory corresponding to the hash value (the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11). The entry containing the partial key is read in order to obtain this information. Again, Fig. 13a illustrates that this can be applied to the partial keys if the original key is greater than 64 bits;

Brandin further teaches executing a conflict check by comparing the confirmer of a partial key derived from the confirmer of an incoming full key with the confirmer of a partial key stored in the memory ((col. 2, line 66 through col. 3, line 11) – the first confirmer (derived from the first partial key of the full key) is compared with a stored first confirmer at the first address). He fails to teach however, actually comparing the keys (in contrast he teaches comparing the values of hashing results produced by applying the transform generator to the keys).

Biran however teaches a system for address protection using a hardware-defined application key, which in fact directly compares the keys in order to mitigate hashing

conflicts (col. 2, lines 58-67 – Biran teaches eliminating the possibility of conflicts occurring by directly comparing the keys (in contrast to Brandin's system of comparing the hashed values of the keys)).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Biran's address protection system using a hardware-defined application key in his own system. By including Biran's method of comparing the keys, rather than comparing the translated keys, Brandin would be able to compare keys that correspond uniquely to the appropriate hardware address, hence eliminating the possibility of hashing conflicts. This system could easily be implemented in hardware (i.e. Brandin's controller which is used to compare the translated keys), while minimizing processing overhead – col. 2, lines 58-67).

As for claim 15, Brandin teaches a hashing apparatus comprising:

means for storing a plurality of partial keys in memory (Fig 13a illustrates two partial keys (elements 316 and 320). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). The keys (which are split into partial keys as illustrated in Fig. 13a-b), are provided to the transform generator – col. 2, lines 54-57). Referring to Fig. 9, each key is stored as an entry in the memory table (i.e. 10[exp]8 keys);

means for applying a hash function to an original key to generate a hash value, the hash function comprising any N bit polynomial (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer

information (i.e. hash value) for the key – This procedure is applied to partial keys in Fig 13a.);

means for accessing the memory according to the hash value, wherein a position to save comprises any N consecutive bits (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49. Referring to Fig. 13a, the original key (element 312) is comprised of two partial keys (elements 316 and 320). The bits in each partial key are stored in a sequential line (based on the key length), each containing less bits than the original key – col. 7, lines 14-39);

means for reading the partial key from the memory corresponding to the hash value, wherein a size to save comprises (less than or equal to) N bits (the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11. The entry containing the partial key is read in order to obtain this information. Again, Fig. 13a illustrates that this can be applied to the partial keys if the original key is greater than 64 bits);

Brandin further teaches executing a conflict check by comparing the confirmer of a partial key derived from the confirmer of an incoming full key with the confirmer of a partial key stored in the memory ((col. 2, line 66 through col. 3, line 11) – the first confirmer (derived from the first partial key of the full key) is compared with a stored first confirmer at the first address). He also teaches the hash table size as 2(exp)N (Fig. 2, 16 entries are disclosed). He fails to teach however, actually comparing the

keys (in contrast he teaches comparing the values of hashing results produced by applying the transform generator to the keys).

Biran however teaches a system for address protection using a hardware-defined application key, which in fact directly compares the keys in order to mitigate hashing conflicts (col. 2, lines 58-67 – Biran teaches eliminating the possibility of conflicts occurring by directly comparing the keys (in contrast to Brandin's system of comparing the hashed values of the keys)).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Biran's address protection system using a hardware-defined application key in his own system. By including Biran's method of comparing the keys, rather than comparing the translated keys, Brandin would be able to compare keys that correspond uniquely to the appropriate hardware address, hence eliminating the possibility of hashing conflicts. This system could easily be implemented in hardware (i.e. Brandin's controller which is used to compare the translated keys), while minimizing processing overhead – col. 2, lines 58-67).

As for claims 2 and 16, Brandin teaches the method of Claim 1 (and apparatus of claim 15), wherein saved bits comprise a consecutive, sequential string of bits, less than or equal to N, which is part of the original key (referring to Fig. 13a, the original key (element 312) is comprised of two partial keys (elements 316 and 320). The bits in each partial key are stored in a sequential line (based on the key length), each containing less bits than the original key – col. 7, lines 14-39).

As for claims 3 and 17, Brandin teaches the method of Claim 2 (and apparatus of claim 16), wherein the partial key comprises a number of bits equal to or more than the number of bits of the original key minus the number of bits of the hash value (referring again to Fig. 13a, partial key A (element 316) is input into the LFSR to generate a hash value (transform) which is equal in size to the partial key. Since the partial key is half the original key's size, the partial key is equal to the size of the original key minus the hash value – col. 7, lines 14-39).

As for claims 4 and 18, Brandin teaches the method of Claim 1 (and apparatus of claim 15), wherein the hash function is implemented by a linear feedback shift register (Fig. 12, element 312 illustrates the LFSR – col. 7, lines 9-11).

8. Claims 6 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combined teachings of Brandin and Biran as applied to claims 1 and 15 above, and in further view of Ji (US PG Publication 2005/0086363 A1).

As for claims 6 and 20, Brandin teaches the method of Claim 1 (and apparatus of claim 15), further comprising:

reading a result from the memory corresponding to the hash value (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49, and the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11).

Brandin fails however to teach forwarding a packet of data according to the result read from the memory.

Ji however teaches a traffic flow management system through a multipath network, which uses a router to forward packets of data. The packets are forwarded in accordance with the information provided to system based on the hash value of the data being forwarded (paragraph 0026, lines 15-20).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Ji's traffic flow management system in order for Brandin to send information referenced by his memory store, as a series of packets. By doing so, Brandin would be able to more efficiently send data referenced by the memory store data, which would in turn improve the load balancing during data transmission (paragraph 008, lines 1-16).

9. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin as applied to claim 7 above, and in further view of Bryg et al., hereinafter Bryg (US Patent 6,430,670 B1).

As for claim 13, Brandin fails to teach the hashing apparatus of claim 7 further including a reverse function generator coupled to the memory wherein the reverse function generator generates the original key based on the partial key and hash value.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key) and tag are used to uniquely identify the original translation of the key. This procedure can be reversed by applying the reverse

hash function on the hash result and the hash identifiers – col. 8, lines 4-21. Note the hash generator hardware is coupled to the system's memory (Fig. 8, element 131).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Bryg's apparatus and method for a virtual hashed page table. By doing so, Brandin would benefit from Bryg's virtual hash translating by utilizing two unique address spaces (either multiple or single hashed page table method) – col. 1, lines 18-28. Bryg's apparatus would provide Brandin with a single architectural virtual hash page table, which supports both methods of virtual addressing. In turn Brandin would benefit by increasing the number of operating systems capable of managing the information, and more efficiently utilize the structure, which in the end would save the end user time and memory as taught by Bryg in col. 2, lines 28-40.

10. Claims 5 and 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over the combined teachings of Brandin and Biran as applied to claims 1 and 15 above, and in further view of Bryg et al., hereinafter Bryg (US Patent 6,430,670 B1).

As for claims 5 and 19, Brandin fails to teach the method of Claim 1 (and apparatus of claim 15), further comprising applying a reverse function on the partial key and hash value to generate the original key.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key) and tag are used to uniquely identify the original translation of the key. This procedure can

be reversed by applying the reverse hash function on the hash result and the hash identifiers – col. 8, lines 4-21.

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Bryg's apparatus and method for a virtual hashed page table. By doing so, Brandin would benefit from Bryg's virtual hash translating by utilizing two unique address spaces (either multiple or single hashed page table method) – col. 1, lines 18-28. Bryg's apparatus would provide Brandin with a single architectural virtual hash page table, which supports both methods of virtual addressing. In turn Brandin would benefit by increasing the number of operating systems capable of managing the information, and more efficiently utilize the structure, which in the end would save the end user time and memory as taught by Bryg in col. 2, lines 28-40.

11. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin as applied to claim 7 above, and in further view of Ji.

As for claim 14, Brandin fails to teach the hashing apparatus of claim 7 further comprising a forwarding engine coupled to the memory, wherein the forwarding engine forwards a data packet according to information read from the memory at an address corresponding to the partial key.

Ji however teaches a traffic flow management system through a multipath network, which uses a router to forward packets of data. The packets are forwarded in accordance with the information provided to system based on the hash value of the data being forwarded (paragraph 0026, lines 15-20).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Ji's traffic flow management system in order for Brandin to send information referenced by his memory store, as a series of packets. By doing so, Brandin would be able to more efficiently send data referenced by the memory store data, which would in turn improve the load balancing during data transmission (paragraph 008, lines 1-16).

12. Claims 21-24 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin in further view of Biran and Bryg.

As for claim 21, Brandin teaches a computer-readable medium having stored thereon instructions for hashing, comprising:

storing a plurality of partial keys in memory (Fig 13a illustrates two partial keys (elements 316 and 320). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). The keys (which are split into partial keys as illustrated in Fig. 13a-b), are provided to the transform generator – col. 2, lines 54-57). Referring to Fig. 9, each key is stored as an entry in the memory table (i.e. 10[exp]8 keys);

applying a hash function to an original key to generate a hash value (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to partial keys in Fig 13a.);

accessing the memory according to the hash value (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49);

reading the partial key from the memory corresponding to the hash value (the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11). The entry containing the partial key is read in order to obtain this information. Again, Fig. 13a illustrates that this can be applied to the partial keys if the original key is greater than 64 bits;

Brandin further teaches executing a conflict check by comparing the confirmer of a partial key derived from the confirmer of an incoming full key with the confirmer of a partial key stored in the memory ((col. 2, line 66 through col. 3, line 11) – the first confirmer (derived from the first partial key of the full key) is compared with a stored first confirmer at the first address). He fails to teach however, actually comparing the keys (in contrast he teaches comparing the values of hashing results produced by applying the transform generator to the keys).

Biran however teaches a system for address protection using a hardware-defined application key, which in fact directly compares the keys in order to mitigate hashing conflicts (col. 2, lines 58-67 – Biran teaches eliminating the possibility of conflicts occurring by directly comparing the keys (in contrast to Brandin's system of comparing the hashed values of the keys)).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Biran's address protection system using a hardware-defined application key in his own system. By including Biran's method of comparing the keys, rather than comparing the translated keys, Brandin would be able to compare keys that correspond uniquely to the appropriate hardware address, hence eliminating the possibility of hashing conflicts. This system could easily be implemented in hardware (i.e. Brandin's controller which is used to compare the translated keys), while minimizing processing overhead – col. 2, lines 58-67).

Brandin further fails to disclose applying a reverse function on the partial key and hash value to generate the original key.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key and tag are used to uniquely identify the original translation of the key. This procedure can be reversed by applying the reverse hash function on the hash result and the hash identifiers – col. 8, lines 4-21). Note the hash generator hardware is coupled to the system's memory (Fig. 8, element 131).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Bryg's apparatus and method for a virtual hashed page table. By doing so, Brandin would benefit from Bryg's virtual hash translating by utilizing two unique address spaces (either multiple or single hashed page table method) – col. 1, lines 18-28. Bryg's apparatus would provide Brandin with a single architectural virtual hash page table, which supports both methods of virtual

addressing. In turn Brandin would benefit by increasing the number of operating systems capable of managing the information, and more efficiently utilize the structure, which in the end would save the end user time and memory as taught by Bryg in col. 2, lines 28-40.

As for claim 22, Brandin teaches the medium of Claim 21, wherein saved bits comprise a consecutive, sequential string of bits, which is part of the original key (referring to Fig. 13a, the original key (element 312) is comprised of two partial keys (elements 316 and 320). The bits in each partial key are stored in a sequential line (based on the key length), each containing less bits than the original key – col. 7, lines 14-39).

As for claim 23, Brandin teaches the medium of Claim 22, wherein the partial key comprises a number of bits equal to larger than the number of bits of the original key minus the number of bits of the hash value (referring again to Fig. 13a, partial key A (element 316) is input into the LFSR to generate a hash value (transform) which is equal in size to the partial key. Since the partial key is half the original key's size, the partial key is equal to the size of the original key minus the hash value – col. 7, lines 14-39).

As for claim 24, Brandin teaches the medium of Claim 21, wherein the hash function is implemented by a linear feedback shift register (Fig. 12, element 312 illustrates the LFSR – col. 7, lines 9-11).

As for claim 26, Brandin teaches a method for accessing data, comprising:

storing a plurality of partial keys in memory (Fig 13a illustrates two partial keys (elements 316 and 320). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). The keys (which are split into partial keys as illustrated in Fig. 13a-b), are provided to the transform generator – col. 2, lines 54-57). Referring to Fig. 9, each key is stored as an entry in the memory table (i.e. 10[exp]8 keys);

applying a function to an original key to generate a value (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to partial keys in Fig 13a.);

accessing the memory according to the value (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49);

reading the partial key from the memory corresponding to the value (the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11). The entry containing the partial key is read in order to obtain this information. Again, Fig. 13a illustrates that this can be applied to the partial keys if the original key is greater than 64 bits;

Brandin further teaches executing a conflict check by comparing the confirmer of a partial key derived from the confirmer of an incoming full key with the confirmer of

a partial key stored in the memory ((col. 2, line 66 through col. 3, line 11) – the first confirmer (derived from the first partial key of the full key) is compared with a stored first confirmer at the first address). He fails to teach however, actually comparing the keys in order to determine which data is accessed (in contrast he teaches comparing the values of hashing results produced by applying the transform generator to the keys).

Biran however teaches a system for address protection using a hardware-defined application key, which in fact directly compares the keys in order to mitigate hashing conflicts (col. 2, lines 58-67 – Biran teaches eliminating the possibility of conflicts occurring by directly comparing the keys (in contrast to Brandin's system of comparing the hashed values of the keys)).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Biran's address protection system using a hardware-defined application key in his own system. By including Biran's method of comparing the keys, rather than comparing the translated keys, Brandin would be able to compare keys that correspond uniquely to the appropriate hardware address, hence eliminating the possibility of hashing conflicts. This system could easily be implemented in hardware (i.e. Brandin's controller which is used to compare the translated keys), while minimizing processing overhead – col. 2, lines 58-67).

Brandin further fails to disclose applying a reverse function on the partial key and hash value to generate the original key.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key and tag are used to uniquely identify the original translation of the key. This procedure can be reversed by applying the reverse hash function on the hash result and the hash identifiers – col. 8, lines 4-21). Note the hash generator hardware is coupled to the system's memory (Fig. 8, element 131).

Again, It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Bryg's apparatus and method for a virtual hashed page table. By doing so, Brandin would benefit from Bryg's virtual hash translating by utilizing two unique address spaces (either multiple or single hashed page table method) – col. 1, lines 18-28. Bryg's apparatus would provide Brandin with a single architectural virtual hash page table, which supports both methods of virtual addressing. In turn Brandin would benefit by increasing the number of operating systems capable of managing the information, and more efficiently utilize the structure, which in the end would save the end user time and memory as taught by Bryg in col. 2, lines 28-40.

13. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over the combined teachings Brandin, Bryg and Biran as applied to claim 21 above, and in further view of Ji.

As for claim 25, Brandin teaches the medium of Claim 21, further comprising:

reading a result from the memory corresponding to the hash value (the address and confirmer information (i.e. hash value) is used to locate the data in

the memory – col. 2, lines 47-49, and the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11).

Brandin fails however to teach forwarding a packet of data according to the result read from the memory.

Ji however teaches a traffic flow management system through a multipath network, which uses a router to forward packets of data. The packets are forwarded in accordance with the information provided to system based on the hash value of the data being forwarded (paragraph 0026, lines 15-20).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Ji's traffic flow management system in order for Brandin to send information referenced by his memory store, as a series of packets. By doing so, Brandin would be able to more efficiently send data referenced by the memory store data, which would in turn improve the load balancing during data transmission (paragraph 008, lines 1-16).

Conclusion

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Sowa et al. (US Patent 6594665 B1) teaches storing hashed values of data in media to allow faster searches and comparison of data.

Horita et al. (US PG Publication 2002/0152389 A1) teaches a distributed digital signature generation method and digitally signed digital document generation method and apparatus.

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Craig E. Walter whose telephone number is (571) 272-8154. The examiner can normally be reached on 8:30a - 5:00p M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mano Padmanabhan can be reached on (571) 272-4210. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Craig E Walter
Examiner
Art Unit 2188

CEW

Reginald G. Bragdon

REGINALD G. BRAGDON
PRIMARY EXAMINER